Opérateur DEFI_MAILLAGE

1 But

Définir un maillage à l'aide de macro-éléments.

Cette commande permet de définir un nouveau maillage à partir de macro-éléments statiques ou dynamiques. Ce nouveau maillage (ne contenant que les supports géométriques des macro-éléments) peut ensuite être "assemblé" à un autre maillage (contenant par exemple des mailles "classiques" grâce à la commande ASSE_MAILLAGE [U4.23.03] et l'option propre à la sous-structuration.

Produit une structure de données de type maillage.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2 Syntaxe

```
ma (maillage) = DEFI_MAILLAGE (

    ♦   DEFI_SUPER_MAILLE = (_F (
        ♦   MACR_ELEM =      l_se    ,                   [l_macr_elem_*]
        ◊   SUPER_MAILLE =          l_mail ,             [l_maille]
        ◊   |  ◊   TRAN      =  /   (tx, ty), ou (tx, ty, tz), [l_R]
                                /   (0.,0.) ou (0.,0.,0.), [DEFAUT]
            |  ◊   ANGL_NAUT =  /   (a),  ou (a, b, c),    [l_R]
                                /   (0.), ou (0.,0.,0.),   [DEFAUT]
               ◊   CENTRE    =  /   (px,py) ou (px,py,pz), [l_R]
                                /   (0.,0.), ou (0.,0.,0.),[DEFAUT]
                      ),),
    ◊   |   RECO_GLOBAL  = (_F  (
            ♦  /  TOUT         =    'OUI' ,
               /  SUPER_MAILLE =    l_maille  ,          [l_maille]
            ◊  |  CRITERE   =  /    'ABSOLU'  ,
                               /    'RELATIF' ,          [DEFAUT]
               |  PRECISION =  /    prec      ,          [R]
                               /    1.D-3     ,          [DEFAUT]
                         ),  ),
        |   RECO_SUPER_MAILLE = (_F (
            ♦   SUPER_MAILLE     =  l_mail ,             [l_maille]
            ♦   GROUP_NO         =  l_gno  ,             [l_group_no]
            ◊   /  OPTION =  'GEOMETRIQUE',              [DEFAUT]
                   ◊  |  CRITERE   =  /  'ABSOLU'  ,
                                      /  'RELATIF' ,     [DEFAUT]
                      |  PRECISION =  /  prec   ,        [R]
                                      /  1.D-3  ,        [DEFAUT]
                /  OPTION =  'NOEUD_A_NOEUD' ,
                /  OPTION =  'INVERSE' ,
                         ),),
    ◊   DEFI_NOEUD = _F  (
            /  ♦  TOUT    =       'OUI'  ,
               ◊  PREFIXE =       pref   ,               [Kn]
               ♦  INDEX   =       (dm,fm,dn,fn),         [l_I]
            /  ♦  NOEUD_FIN =  no_fin ,                  [noeud]
               ♦  SUPER_MAILLE =  mail   ,               [maille]
               ♦  NOEUD_INIT   =  no_ini ,               [noeud]
                      ),),
    ◊   DEFI_GROUP_NO = _F  (
            /  ♦  /  TOUT       =     'OUI'  ,
                  /  SUPER_MAILLE =     mail   ,         [maille]
               ◊  PREFIXE   =     pref   ,               [Kn]
               ♦  INDEX     =     (dm,fm,dn,fn),         [l_I]
            /  ♦  GROUP_NO_FIN =  gno_fin,               [group_no]
               ♦  SUPER_MAILLE =  mail   ,               [maille]
               ♦  GROUP_NO_INIT=  gno_ini,               [group_no]
                      ),),
                                  )
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3 Généralités

Dans la documentation de cette commande, on parlera de :

- macro-élément : objet de type `macr_elem_stat` ou `macr_elem_dyna`,
- super-maille : entité géométrique supportant un macro_élément,
- maillage **initial** quand on désigne le maillage qui a servi à engendrer un macro-élément,
- maillage **final** pour désigner le maillage produit par cette commande.

Par extension ces adjectifs **initial/final** s'appliqueront aux entités attachées aux maillages : nœud, maille, groupe de nœuds.

Pratiquement, pour construire le maillage final :

- on définit des super-mailles en positionnant dans l'espace (2D ou 3D) des macro-éléments existants (un même macro-élément peut engendrer plusieurs super-mailles),
- on recolle les super-mailles entre elles,
- on renomme, si on le veut, certains nœuds,
- on crée, si on le veut, certains groupes de nœuds.

Remarques :

On peut constater que le maillage créé par cette commande n'est formé que de super-mailles. On ne peut donc pas (par exemple), le dessiner avec les post-processeurs usuels. Des possibilités d'y remédier pourront exister avec la commande `DEFI_SQUELETTE` [U4.24.01].

Pour mélanger des éléments finis "classiques" et des sous-structures, il faut utiliser l'opérateur de "concaténation" de maillages [U4.23.03] : `mag = ASSE_MAILLAGE ( MAILLAGE= (m1, m2))`

Un maillage résultant de l'opérateur `DEFI_MAILLAGE` contient :

- des super-mailles,
- des nœuds,
- des groupes de nœuds.

Les super-mailles sont définies par translation/rotation de macro-éléments.

Comme une maille "classique", une super-maille est entièrement définie par la liste de ses nœuds. Les coordonnées des nœuds des mailles sont celles des nœuds externes des macro-éléments transformées par la transformation géométrique : translation, rotation ...

Si on n'effectue pas de recollement (cf. `RECO_GLOBAL` / `RECO_SUPER_MAILLE`), le maillage a autant de nœuds que la somme des nœuds des super-mailles.

Convention C1 :

Lorsque l'on "recolle" les super-mailles, on élimine certains nœuds. Par convention, lors d'une élimination de nœuds coïncidants, on conserve le nœud (et donc ses coordonnées) qui provient de la première maille de la liste `l_mail` (cf. `RECO_GLOBAL` / `RECO_SUPER_MAILLE`).

Comme dans tout maillage *Aster*, les nœuds sont **nommés**. Par défaut, les noms des nœuds sont donnés par le programme sous la forme : `Nijk` où `ijk` est un numéro compris entre `1` et `999999.9`.

Les mots clé `DEFI_NOEUD` et `DEFI_GROUP_NO` permettent à l'utilisateur de renommer certains nœuds et de définir des groupes de nœuds.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4 Opérandes

4.1 Mot clé DEFI_SUPER_MAILLE

♦ `DEFI_SUPER_MAILLE =`

Ce mot clé facteur permet de définir les super-mailles du maillage à l'aide des macro-éléments.

4.1.1 Opérande MACR_ELEM

♦ `MACR_ELEM = l_se`

`l_se` est la liste des noms des macro-éléments qui vont engendrer les mailles.

4.1.2 Opérande SUPER_MAILLE

◊ `SUPER_MAILLE = l_mail`

`l_mail` est la liste des noms que l'on veut donner aux mailles. Cet argument est facultatif. En son absence, on donnera aux mailles les noms des macro-éléments (ceci est évidemment impossible si on veut utiliser plusieurs fois le même macro-élément).

4.1.3 Opérandes de transformations géométriques

◊ | ◊ `TRAN =`

Ce mot clé définit la translation à appliquer au macro-élément:

- si on est en **2D**, on attend 2 réels : (`tx, ty`),
- si on est en **3D**, on attend 3 réels : (`tx, ty, tz`).

| ◊ `ANGL_NAUT =`

◊ `CENTRE =`

Ces mots clé définissent la rotation à appliquer au macro-élément.

Si on est en **2D**, on attend 3 réels :

- `a` est l'angle (en degrés) de rotation dans le plan pour `ANGL_NAUT`,
- `px` et `py` sont les coordonnées du centre de rotation pour `CENTRE`.

Si on est en **3D**, on attend 6 réels :

- `a, b, c` sont les angles nautiques (α, β, γ) de la rotation (en degrés). (Cf. l'opérateur `AFFE_CARA_ELEM` [U4.42.01]) pour `ANGL_NAUT`,
- `px`, `py` et `pz` sont les coordonnées du centre de rotation pour `CENTRE`.

Remarque importante :

*On sait que l'ordre des mots clés n'est pas significatif pour Aster. L'opération de translation/rotation est **conventionnellement** faite dans l'ordre rotation **puis** translation. Ces deux opérations ne commutent pas en général.*

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.2 Mot clé RECO_GLOBAL

```
◊  |  RECO_GLOBAL =
   ♦  /  TOUT         = 'OUI' ,
      /  SUPER_MAILLE = l_maille,
   ◊  |  CRITERE   =  /   'ABSOLU'  ,
                      /   'RELATIF' ,              [DEFAUT]
      |  PRECISION =  /   prec,
                      /   1.D-3,                   [DEFAUT]
```

Ce mot clé permet de recoller **automatiquement** un ensemble de super-mailles (désignées par le mot clé `SUPER_MAILLE` ou le mot clé `TOUT`) avec un critère de proximité géométrique : 2 nœuds de 2 super-mailles différentes `m1` et `m2` seront confondus si la distance qui les sépare est :

```
<  prec                     (CRITERE = 'ABSOLU'),
<  prec*min(d(m1),d(m2))    (CRITERE = 'RELATIF').
```

où `d(mi)` note la plus petite distance entre 2 nœuds de la super-maille `mi`.

Remarques :

Deux nœuds d'une même maille ne seront jamais recollés.

Si une maille ne contient qu'un seul nœud, il faut utiliser le `CRITERE = 'ABSOLU'`*.*

4.3 Mot clé RECO_SUPER_MAILLE

◊ `RECO_SUPER_MAILLE =`

Ce mot clé facteur permet de recoller "**à la main**" certaines super-mailles désignées par l'utilisateur. Les super-mailles que l'on peut recoller sont celles qui ont été définies par le mot clé `DEFI_SUPER_MAILLE`.On recolle alors les super-mailles via des groupes de nœuds. Pour dire ce que l'on veut recoller il faut donc donner des couples (maille, groupe de nœuds (du maillage initial)).

Remarques :

Lorsqu'on donne un couple (maille , groupe de nœuds), on désigne la liste des nœuds du groupe de nœuds qui sont externes pour le macro-élément qui définit la super-maille. C'est en fait l'intersection du groupe de nœuds et du bord de la sous-structure. Cette liste est ordonnée comme le groupe de nœuds initial.

En principe, lorsqu'on recolle 2 mailles via 2 groupes de nœuds, l'ensemble des nœuds désignés doit se recoller (cf. la convention choisie par le mot clé `OPTION` *). Un message d'alarme sera émis si ce n'est pas le cas.*

4.3.1 Opérandes SUPER_MAILLE / GROUP_NO

◊ `SUPER_MAILLE =`

On donne ici la liste des mailles à recoller. En général, on recolle les mailles 2 par 2.

Pour les "coins", il peut être agréable de recoller toutes les mailles concourantes en une seule fois (par exemple les 4 super-cubes qui se partagent la même arête).

◊ `GROUP_NO =`

On donne ici la liste des groupes de nœuds à recoller. Cette liste est de même longueur que la liste des mailles.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.3.2 Opérande OPTION

◊ OPTION =

Ce mot permet de choisir la convention de recollement des listes de nœuds définis par les groupes de nœuds.

- 'GEOMETRIQUE' :

  Le programme va confondre les nœuds par des considérations de proximité géométrique. (Cf. mot clé : RECO_GLOBAL)

- 'NOEUD_A_NOEUD' / 'INVERSE' :

Soit : $G1=\{A1, B1, C1\}$

$G2=\{A2, B2, C2\}$

$G3=\{A3, B3, C3\}$

Si OPTION = 'NOEUD_A_NOEUD' , GROUP_NO = (G1, G2, G3)

on va recoller : **A1** avec **A2** avec **A3**

B1 avec **B2** avec **B3**

C1 avec **C2** avec **C3**

Si OPTION = 'INVERSE' , GROUP_NO = (G1, G2, G3)

on va recoller : **C1** avec **A2** avec **A3**

B1 avec **B2** avec **B3**

A1 avec **C2** avec **C3**

Attention :

Pour option 'INVERSE' , seul le premier groupe de nœuds de la liste des GROUP_NO est "retourné".

4.4 Mot clé DEFI_NOEUD

◊ DEFI_NOEUD =

Ce mot clé facteur permet de renommer tout ou partie des nœuds du maillage.

4.4.1 Opérandes TOUT / PREFIXE / INDEX

```
| ♦ TOUT = 'OUI' ,
  ◊ PREFIXE = pref,
  ♦ INDEX = (dm, fm, dn, fn),
```

Ces mots clé permettent de renommer tous les nœuds du maillage. La convention de renommage est la suivante (en pseudo FORTRAN) :

```
no_fin (K8) = pref//no_mail(dm:fm)//no_ini(dn:fn)
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Ce qui veut dire que le nom d'un nœud sera formé en concaténant :

- le préfixe éventuellement donné par l'utilisateur,
- une sous-chaine de caractères extraite du nom de la maille qui porte ce nœud (cf. la convention C1 d'élimination des nœuds énoncée ci-dessus [§ 3]). On prend les caractères de rang compris entre `dm` et `fm`. Si `dm > fm`, cette sous-chaine est vide,
- une sous-chaine de caractères extraite du nom du nœud (dans son maillage initial). On prend les caractères de rang compris entre `dn` et `fn`. Si `dn > fn`, cette sous-chaîne est vide.

Il faut donc que : `ltot = longueur(préfixe) + (fm-dm+1) + (fn-dn+1) ▪ 8`

On rappelle que 2 nœuds ne peuvent avoir le même nom dans un même maillage. Le but du "jeu" pour l'utilisateur est d'arriver à renommer certains nœuds (sans trop d'efforts de sa part) de manière conventionnelle sans que cette convention conduise à des noms identiques.

Un cas fréquent est le suivant :

si les maillages qui ont donné naissance aux macro-éléments proviennent d'un pré-processeur qui engendre des noms de nœuds de la forme `NOijklmn` et si l'utilisateur donne à ses super-mailles des noms à 2 caractères : `SA`, `SB`,.... la séquence :

```
DEFI_NOEUD = _F ( TOUT= 'OUI' , INDEX=(1, 2, 3, 8, ))
```

engendrera des nœuds de noms : `SA000001, SA000002,... , SB000001,`

4.4.2 Opérandes NOEUD_FIN / SUPER_MAILLE / NOEUD_INIT

```
|  ♦  NOEUD_FIN    =     no_fin,
   ♦  SUPER_MAILLE =     mail,
   ♦  NOEUD_INIT   =     no_ini,
```

Ces mots clés permettent de renommer des nœuds **un par un** :

- `no_fin` est le nom que l'on veut donner au nœud du maillage que l'on crée (final).
- `mail` et `no_ini` identifient le nœud à renommer : `mail` est le nom de la super-maille qui porte le nœud, `no_ini` est le nom du nœud dans le maillage qui a servi à créer le macro-élément qui a défini la super-maille `mail`.

4.5 Mot clé DEFI_GROUP_NO

◊ `DEFI_GROUP_NO =`

Ce paragraphe est presque identique au précédent (`DEFI_NOEUD`) en remplaçant le mot `NOEUD_` par le mot `GROUP_NO`.

Ce mot clé facteur permet de définir des groupes de nœuds à partir de groupes existant dans les maillages initiaux des macro-éléments.

Remarque :

Un groupe de nœuds initial peut contenir des nœuds qui n'appartiennent pas aux bords des macro-éléments. Ces nœuds internes n'existent donc pas dans le maillage final. Par commodité, on prend la convention de créer quand même le groupe réduit à son intersection avec le bord du macro-élément.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.5.1 Opérandes TOUT / SUPER_MAILLE / PREFIXE / INDEX

```
| ♦ / TOUT         = 'OUI' ,
    / SUPER_MAILLE = mail,
  ◊ PREFIXE = pref,
  ♦ INDEX   = (dm, fm, dn, fn),
```

Ces mots clés permettent de créer tous les groupes de nœuds correspondants aux groupes du maillage initial associé à la maille `mail` ou à toutes les mailles si :

`TOUT= 'OUI'`.

La convention de renommage est la suivante (en pseudo FORTRAN) :

```
gno_fin(k8) = pref//no_mail(dm:fm)//gno_ini(dn:fn)
```

Ce qui veut dire que le nom d'un groupe de nœuds sera formé en concaténant :

- le préfixe éventuellement donné par l'utilisateur,
- une sous-chaine de caractères extraite du nom de la maille,
- une sous-chaine de caractères extraite du nom du `group_no` du maillage initial.

Il faut donc que :

```
ltot= longueur(préfixe) + (fm-dm+1) + (fn-dn+1) ▪8
```

Un cas fréquent est le suivant : les maillages qui ont donné naissance aux macro-éléments proviennent d'un pré-processeur qui engendre des noms de la forme `GRNOijkl`. Si l'utilisateur donne à ses super-mailles des noms à 2 caractères : `SA`, `SB`, ..., la séquence :

```
DEFI_GROUP_NO=_F(TOUT= 'OUI' , PREFIXE='GN' , INDEX=(1,2,5,8))
```

Engendrera des groupes de nœuds de noms :

```
GNSA0001, GNSA0002,... , GNSB0001.
```

4.5.2 Opérandes GROUP_NO_FIN / SUPER_MAILLE / GROUP_NO_INIT

```
| ♦ GROUP_NO_FIN  = gno_fin,
  ♦ SUPER_MAILLE  = mail,
  ♦ GROUP_NO_INIT = gno_ini,
```

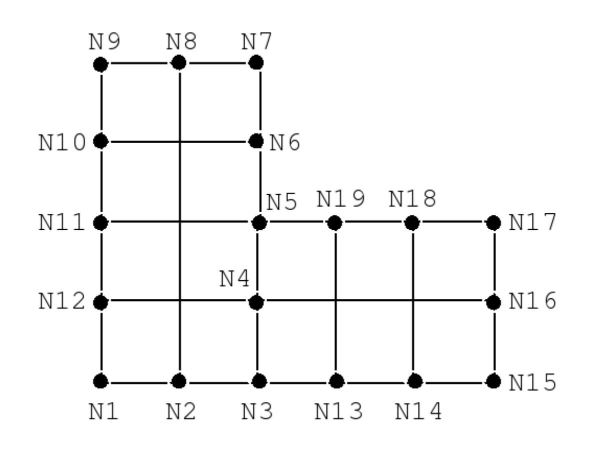
Ces mots clés permettent de créer des groupes de nœuds **un par un** :

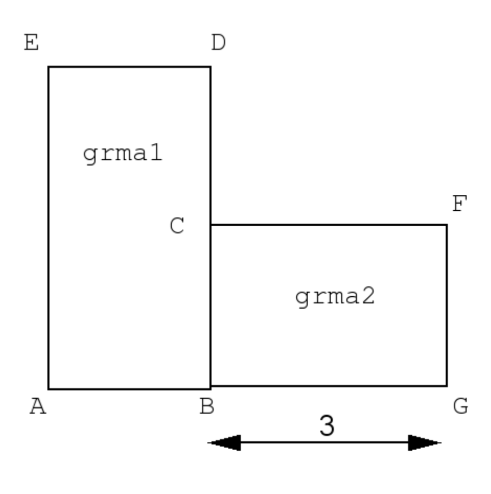
- `gno_fin` est le nom que l'on veut donner au `GROUP_NO`,
- `mail` et `gno_ini` identifient le `GROUP_NO` initial :
  - `mail` est le nom de la super-maille qui porte le `GROUP_NO`,
  - `gno_ini` est le nom du `GROUP_NO` du maillage initial.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

5 Exemple

Soit le maillage `m1` :






```
GROUP_NO :

AB = (N1 N2 N3)
BC = (N3 N4 N5)
CD = .....
......
```

```
GROUP_MA :

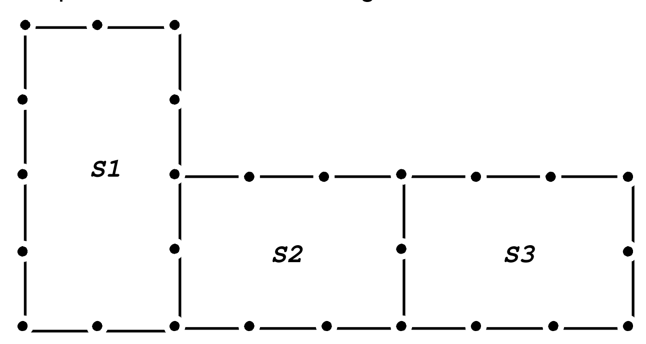
grma1
grma2
```

Sur ce maillage `m1` on définit 2 `macr_elem_stat`.

```
mo1 =  AFFE_MODELE      (  AFFE = _F (GROUP_MA = grma1)...)
mo2 =  AFFE_MODELE      (  AFFE = _F (GROUP_MA = grma2)...)

S1 =   MACR_ELEM_STAT   (  DEFINITION = _F  (MODELE = mo1...)
                              EXTERIEUR  = _F  (GROUP_NO = (AB,BC,CD,DE,EA))
                           ...)
S2 =   MACR_ELEM_STAT   (  DEFINITION = _F  (MODELE = mo2...)
                              EXTERIEUR  = _F  (GROUP_NO = (BC,BG,FG,CF))
                           ...)
```

On peut alors définir le maillage `m2` :



Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
m2 =   DEFI_MAILLAGE(

    DEFI_SUPER_MAILLE=(
        _F(MACR_ELEM = S1)  ,
        _F(MACR_ELEM = S2   ,   SUPER_MAILLE = S2,  ) ,
        _F(MACR_ELEM = S2   ,   SUPER_MAILLE = S3,  TRAN = 3.),),

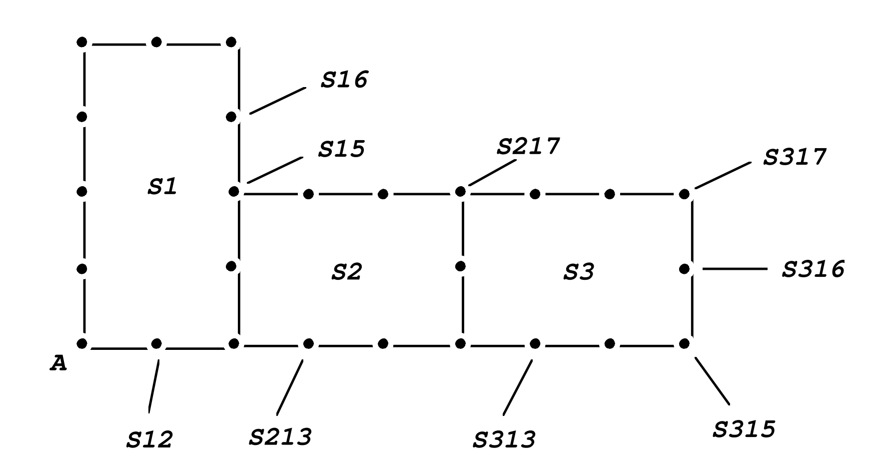
    RECO_SUPER_MAILLE=(
        _F(SUPER_MAILLE=(S1,S2),GROUP_NO=(BC,BC),OPTION='NOEUD_A_NOEUD'),
        _F(SUPER_MAILLE=(S2,S3),GROUP_NO=(FG,BC),OPTION='INVERSE'),),

    DEFI_NOEUD=(
        _F(TOUT = 'OUI' ,INDEX = (1, 2, 2, 3)),
        _F(NOEUD_FIN = A,   SUPER_MAILLE = S1 , NOEUD_INIT = N1),),

    DEFI_GROUP_NO =
       _F(GROUP_NO_FIN = FG, SUPER_MAILLE = S3,GROUP_NO_INIT = FG), )
```

Le maillage obtenu contient :

- 3 super-mailles : `S1, S2, S3`
- 26 nœuds : `A, S12, ..., S317`
- 1 `GROUP_NO` : `FG = (S315, S316, S317)`



Remarque :

Le recollement des super-mailles aurait pu être fait plus simplement par :
`RECO_GLOBAL = _F (TOUT = 'OUI') .`

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)